COMMANDES UNIX DE BASE

Avertissement : Unix étant écrit en langage C, il distingue les minuscules des majuscules.

1 Droits d'accès

Dans un système Unix, l'ensemble des utilisateurs est connu de toutes les machines. Cet ensemble est toujours découpé en sous-ensembles, que l'on appelle des groupes. Ces derniers sont déterminés par l'administrateur du système (le super utilisateur, ou root). Tout utilisateur fait partie d'un groupe, au moins.
Chaque fichier (ou répertoire) présent sur le système appartient à l'utilisateur qui l'a créé : le propriétaire. Pour tout fichier (respectivement répertoire), il existe des droits d'accès (ou permissions) modifiables par son propriétaire. Ces droits concernent la lecture (**r**), l'écriture (**w**) et l'exécution dans le cas d'un fichier ou l'exploration dans le cas d'un répertoire (**x**). Ils sont gérables relativement au propriétaire lui-même (**u**), à son groupe (**g**), aux autres utilisateurs (**o**), ou encore à tous les utilisateurs (**a**). Les permissions affectées à un fichier ou un répertoire sont décrites par 3 triplets accolés, le premier se rapportant au propriétaire, le deuxième à son groupe et le troisième aux autres utilisateurs. A la demande de l'utilisateur, Unix peut afficher les informations rattachées à un fichier ou un répertoire, dont les droits d'accès. En voici trois exemples pour l'utilisateur eric, créé le 16 septembre et qui appartient au groupe an1 :

permissions	propriétaire	groupe	taille	date de création		nom
d r w x r - x r - x	eric	an1	4096	sep	16	eric
- r w - r - - r - -	eric	an1	1040	oct	10	toto.c
- r w x r - x r - -	eric	an1	5212	oct	12	toto.exe

Le premier caractère distingue les répertoires (**d**) des fichiers (**-**) et des liens (**l**).
Le répertoire "eric" est visible de tout le monde, mais seul son propriétaire peut y écrire.
Les fichiers "toto.c" et "toto.exe" sont accessibles en lecture par tout le monde, mais ne sont modifiables que par leur propriétaire. Tous les utilisateurs du groupe "an1" peuvent exécuter "toto.exe", tandis que "toto.c" n'est pas exécutable.

2 Administration

man pour afficher la documentation d'une commande unix
passwd pour modifier le mot de passe de l'utilisateur présent
ps -aux pour afficher les processus en cours d'exécution, leur numéro (PID) et leur état
kill -9 pour supprimer un processus en cours, à condition d'en être le propriétaire

3 Manipulations de fichiers

cp pour dupliquer un fichier
mv pour déplacer ou renommer un fichier
cmp pour comparer deux fichiers
diff idem, avec arrêt sur la première différence rencontrée
rm pour détruire un ou plusieurs fichiers
cat pour afficher le contenu d'un petit fichier à l'écran
more idem, mais page par page
less idem, avec retour en arrière possible (on quitte cette commande en tapant q)
gcc pour compiler un fichier écrit en langage C

chmod pour modifier les permissions d'un fichier ou d'un répertoire
grep pour chercher une chaîne de caractères dans un fichier, sans l'éditer
find pour chercher un fichier dans une arborescence
locate idem

4 Manipulations de répertoires

pwd pour afficher le nom du répertoire courant
cd pour changer de répertoire de travail
ls pour afficher la liste des éléments d'un répertoire
ls -l idem avec visualisation des droits d'accès
cp -r pour copier de manière récursive un répertoire
mv pour déplacer ou renommer un répertoire
mkdir pour créer un répertoire vide
rmdir pour détruire un répertoire (à condition qu'il soit vide)
rm -r pour détruire un répertoire qui n'est pas vide (à manipuler avec précaution!)

5 Archivage de répertoires

Afin d'archiver des programmes ou des données, il est utile de regrouper des fichiers et des répertoires tout en conservant leur arborescence. Une commande unique dotée de différents compléments s'applique à l'archivage, c'est la commande **tar**.
tar cvf pour archiver un répertoire
tar xvf pour désarchiver un fichier d'archive
tar tf pour afficher le contenu d'une archive, sans la désarchiver

tar cvfz pour archiver un répertoire et compresser l'archive
tar xvfz pour décompresser puis désarchiver un fichier d'archive

6 Compression de fichiers

Certains fichiers comportent des caractères utiles à leur présentation, mais inutiles quant à leur exploitation par un logiciel. Pour économiser de la place quand on les stocke, éventuellement en vue de leur transfert, on peut les compresser, en supprimant les informations redondantes qu'ils contiennent. Appliquées à un fichier donné, ces commandes créent un nouveau fichier de même nom, mais d'extension différente, dépendante de l'outil utilisé.
compress pour compresser un fichier
uncompress pour décompresser un fichier
Deux utilitaires plus performants prennent le pas sur **compress**. Ils s'appellent **gzip** et **bzip2**.On décompresse respectivement avec **gunzip** et **bunzip**.
L'archivage, suivi d'une compression, est fréquemment utilisé en vue de sauvegarder un ensemble de données stockées sur une machine.

7 Exemples

L'utilisateur eric vient de se connecter sur la machine pg4. Dans une fenêtre terminal, il tape les commandes suivantes, indiquées ici en gras. Les réponses fournies apparaissent en italique.

eric@pg4:~$	affichage qui suit le “login”
pwd	où suis je ?
/home/eric	dans mon “home directory”
cd ..	remontée d'un cran dans l'arborescence,
pwd	car .. désigne le répertoire père
/home	
cd ..	
pwd	
/	on est arrivé à la racine
cd	retour immédiat chez soi
pwd	
/home/eric	
cd /home/yann	un tour chez le voisin
pwd	
/home/yann	
cd ../eric	retour relatif chez soi
pwd	
/home/eric	c'est le répertoire courant
ls	
Documents	contenu du répertoire courant
cmdesUnix	
ls /home	
yann eric ...	contenu du répertoire /home
pwd	sans s'être déplacé pour autant!
/home/eric	
mkdir seance1	création du répertoire
ls seance1	baptisé seance1,
Total 0	qui est vide,
rmdir seance1	suivie de sa destruction
cp -r /stk/res/an1/seance2 rep1	création du répertoire rep1, conformément
ls rep1	au répertoire /stk/res/an1/seance2
prog.c phone	
rm -r rep1	destruction du répertoire non vide
mkdir rep2	création de rep2, vide
cd rep2	rep2 devient le répertoire courant
cp /stk/res/an1/seance2/* .	copie locale de tous les fichiers contenus
ls	dans /stk/res/an1/seance2
prog.c phone	
cp prog.c prog1.c	duplication de prog.c dans prog1.c
cmp prog.c prog1.c	aucune réponse car les deux fichiers sont identiques
mv prog.c premier	changement de nom de prog.c
ls	
premier prog1.c phone	
cat premier	affichage à l'écran de premier, mais on n'y voit rien...

less premier	maintenant on voit tapez q pour quitter
gcc premier	erreur, car le compilateur C n'accepte que les extensions.c
mv premier premier.c	on corrige...
gcc premier.c	maintenant, tout va bien ...
ls	d'où la création, par défaut,
a.out premier.c prog1.c phone	de l'exécutable a.out
rm a.out	attention aux noms par défaut,
gcc -o prim.exe premier.c	il vaut mieux qu'ils soient explicites
gcc -o prim.exe -Wall -g premier.c	deux options souvent utiles -g pour utilisation ultérieure du debugger -Wall pour afficher tous les avertissements détectés
ddd prim.exe	exécution de "prim.exe" sous contrôle du debugger
ls -l	
-rw- r- - r- - premier.c	
-rwx r-x r-x prim.exe	
-rw- r- - r- - prog.c	
chmod o-x prim.exe	interdiction aux utilisateurs autres que ceux de
ls -l prim.exe	mon groupe d'exécuter ce programme
-rwx r-x r- - prim.exe	
rm prog.c prim.exe	il est inutile de conserver ces fichiers
cd ..	remontée d'un cran dans l'arborescence
tar cvf cours2 rep2	archivage du répertoire complet dans cours2
ls	
cours2 Documents rep2	cours2 est un fichier
gzip cours2	
ls	
cours2.gz Documents rep2	il est maintenant compressé
mkdir garder	
mv cours2 garder	
cd garder	
gunzip cours2	
tar xvf cours2	
pwd	
/home/eric/garder	
ls	
rep2	
ps aux	affiche tous les processus en cours d'exécution, affublés de leur numéro et leur état
kill -9 3802	supprime le processus numero 3802
man ls	affiche le manuel d'utilisation de la commande ls